

# Time-Domain Approximation by Iterative Methods

L. E. McBRIDE, JR., MEMBER, IEEE, H. W. SCHAEFGEN, MEMBER, IEEE, AND K. STEIGLITZ, MEMBER, IEEE

**Abstract**—An iterative procedure is presented which permits the determination of a rational transfer function in the Laplace transform variable  $s$  which is optimal with respect to given input and output time-functions. The optimal system of a particular order is defined as the one whose output when subjected to the known input function is nearest in the time integral square sense to the desired output function. The method is thus applicable to a number of problems involving the minimization of an integral square error.

To illustrate the technique, a set of optimal lumped-parameter delay lines is synthesized and their characteristics investigated; the behavior and convergence of the iteration in these problems is also studied.

A comparison of other iterative methods applicable to the same problems leads to the conclusion that the proposed procedure has real advantages in computational simplicity and speed of convergence.

TIME-DOMAIN approximation can be regarded as the problem of finding a network of given complexity which, when subjected to a particular input, will produce (at least approximately) a desired output as a function of time. If it is assumed that any rational function of given order can be synthesized as the transfer function of a network, the problem is reduced to finding the coefficients of a ratio of polynomials in the transform variable,  $s$ , which best fits the required input-output relation. It is in this sense identical to the problem of identifying an unknown linear system from a sample of its input and output functions.

Among the computational procedures proposed for linear system identification, Kalman's [1] method for finding the coefficients of a rational  $z$ -transform function by linear regression is among the easiest to use. However, as a method for time-domain synthesis it has two disadvantages: it results in a sampled-data model which must then be approximated by a rational function of  $s$ ; and it minimizes an artificial error criterion which does not correspond directly to any measurable error. It has been shown [2] that the second objection can be eliminated

Manuscript received August 30, 1965; revised June 17, 1966. This work was partially supported by National Science Foundation Grant GK-270, and Army Research Office Contract DA-31-124-ARO-D292; it made use of computer facilities supported in part by Grant NSF GP-579. An abridged version of this paper was presented at the 1965 Third Allerton Conference on Circuit and System Theory, University of Illinois, Urbana.

L. E. McBride, Jr., is with the Metals and Controls Division, Texas Instruments Incorporated, Attleboro, Mass. He was formerly with the Department of Electrical Engineering, Princeton University.

K. Steiglitz is with the Department of Electrical Engineering, Princeton University, Princeton, N. J.

H. W. Schaeffgen is with the Aeronutronic Division, Philco Corporation, Newport Beach, Calif. He was formerly with the Department of Electrical Engineering, Princeton University.

by using an iterative procedure, with the Kalman estimate as first iteration; if the iteration converges, as it does in many practical cases, the resulting transfer function minimizes the mean square error between desired and actual model output. It is the purpose of this paper to show that similar iterative procedures can be applied directly to continuous system identification and to give examples of their effectiveness in time-domain approximation.

Let the system input be  $x(t)$  and the desired output function  $y(t)$ , both defined for  $t \geq 0$  [ $x(t)$  may be an ideal impulse, in which case  $y(t)$  is the desired network impulse response]. If  $v(t)$  is the actual response of the network to  $x(t)$ , a natural error criterion is

$$J = \int_0^T [v(t) - y(t)]^2 dt = \int_0^T e^2(t) dt. \quad (1)$$

The transfer function of the network to be synthesized has the form

$$\begin{aligned} G(s) &= \frac{a_n + a_{n-1}s + \cdots + a_0s^n}{b_n + b_{n-1}s + \cdots + s^n} \\ &= \frac{a_n s^{-n} + a_{n-1}s^{-n+1} + \cdots + a_0}{b_n s^{-n} + b_{n-1}s^{-n+1} + \cdots + 1} = \frac{N(s)}{D(s)}. \end{aligned} \quad (2)^1$$

(In most problems  $a_0$  is omitted because it is known that the network impulse response should not contain an impulse at  $t = 0$ ; it is also possible to assume a numerator order different from that of the denominator, but this alternative will be ignored to simplify the notation.) Then if  $X(s)$  is the Laplace transform of  $x(t)$ , etc.,

$$V(s) = \frac{N(s)}{D(s)} X(s), \quad E(s) = V(s) - Y(s) \quad (3)$$

and

$$\begin{aligned} \frac{\partial E(s)}{\partial a_i} &= \frac{s^{-i}}{D(s)} X(s) = P_{a_i}(s) \\ \frac{\partial E(s)}{\partial b_i} &= \frac{-s^{-i}N(s)}{D^2(s)} X(s) = -P_{b_i}(s). \end{aligned} \quad (4)$$

Now

$$\begin{aligned} \frac{\partial J}{\partial a_i} &= 2 \int_0^T e(t) \frac{\partial e(t)}{\partial a_i} dt \\ \frac{\partial J}{\partial b_i} &= 2 \int_0^T e(t) \frac{\partial e(t)}{\partial b_i} dt \end{aligned} \quad (5)$$

<sup>1</sup> Division of numerator and denominator by  $s^n$  avoids the necessity for differentiation in the first iteration; it is equivalent to assuming that the initial denominator is  $s^n$ , and has no effect on subsequent iterations.

and, by the definition of the Laplace transform,

$$P_{a_i}(s) = \frac{\partial}{\partial a_i} \int_0^\infty e(t) \exp(-st) dt \\ = \int_0^\infty \frac{\partial e(t)}{\partial a_i} \exp(-st) dt; \quad (6)$$

therefore,

$$p_{a_i}(t) = \frac{\partial e(t)}{\partial a_i}, \quad p_{b_i}(t) = \frac{-\partial e(t)}{\partial b_i}$$

and a necessary condition for the values of  $a_i$  and  $b_i$  to be optimal with respect to the error criterion  $J$  is

$$\frac{\partial J}{\partial a_i} = 2 \int_0^T e(t) p_{a_i}(t) dt = 0 \quad i = 1, \dots, n \\ \frac{\partial J}{\partial b_i} = 2 \int_0^T e(t) p_{b_i}(t) dt = 0 \quad i = 1, \dots, n \quad (7)$$

where  $p_{a_i}$ ,  $p_{b_i}$  may be regarded as the output from a set of filters described by (4).

Unfortunately, (7) is a highly nonlinear set of equations in the coefficients  $a_i$  and  $b_i$ , and no method of direct solution is apparent. Three ways of attacking the problem have been considered; they can be called for convenience Gradient Methods, Mode 1 Iteration, and Mode 2 Iteration.

#### GRADIENT METHODS

Since (7) defines a method for determining the gradient of  $J$  in coefficient space, one straightforward computational method is based on a descent to the minimum. For any initial set of  $a$ 's and  $b$ 's,  $e(t)$ ,  $p_{a_i}(t)$ , and  $p_{b_i}(t)$  can be obtained by simulating the filtering operations of (3) and (4) on either a digital or analog computer. The gradient can then be computed from (7) and indicates in which direction each coefficient must be altered to reduce  $J$ . A suitable small increment is then added to each coefficient, the resulting  $e$ ,  $p$ 's, and gradient computed and the process repeated until the gradient is approximately zero. A variety of more sophisticated programming methods for finding the minimum given a gradient computation procedure is also available [3].

These methods have been tested on similar problems in sampled-data systems and have been found effective. Usually, however, a very large number of iterations is required for convergence, and the procedure is particularly unwieldy when a large number of coefficients (e.g., more than ten) is to be computed; because the other iterative methods seem to work so well for time-domain synthesis, gradient techniques were not tried in this case.

#### MODE 1 ITERATION

A new error,  $e_1(t)$ , is defined such that

$$E_1(s) = \frac{N_j(s)}{D_{j-1}(s)} X(s) - \frac{D_j(s)}{D_{j-1}(s)} Y(s) \quad (8)$$

where the subscript  $j$  refers to the iteration number. It is assumed that the coefficients of  $D_{j-1}$  have been previously computed, and are regarded as fixed during the  $j$ th iteration. Therefore,

$$\frac{\partial E_1(s)}{\partial a_i} = \frac{s^{-i}}{D_{j-1}(s)} X(s) = \hat{P}_{a_i}(s) \quad (9a)$$

$$\frac{\partial E_1(s)}{\partial b_i} = \frac{-s^{-i}}{D_{j-1}(s)} Y(s) = -\hat{P}_{b_i}(s). \quad (9b)$$

Since the expressions in (9) are independent of  $a_i$  and  $b_i$ , and  $e_1(t)$  is a linear function of these coefficients, the gradient of  $J$  is linear and the optimal values of the coefficients can be obtained by linear regression.

If the coefficient vector  $\gamma$  and signal vector  $\mathbf{q}(t)$  are defined by

$$\gamma = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ -b_1 \\ \vdots \\ -b_n \end{bmatrix}, \quad \mathbf{q}(t) = \begin{bmatrix} \hat{p}_{a_1}(t) \\ \vdots \\ \hat{p}_{a_n}(t) \\ \hat{p}_{b_1}(t) \\ \vdots \\ \hat{p}_{b_n}(t) \end{bmatrix},$$

$e_1(t)$  and  $J_1$  can be written as

$$e_1(t) = \gamma' \mathbf{q}(t) - \hat{p}_{b_0}(t) \quad (10)$$

$$J_1 = \int_0^T e_1^2(t) dt,$$

and the gradient of  $J_1$  with respect to  $\gamma$  is

$$\text{grad } J_1 = 2 \left[ \int_0^T \mathbf{q}(t) \mathbf{q}'(t) dt \right] \gamma - 2 \int_0^T \hat{p}_{b_0}(t) \mathbf{q}(t) dt \\ = 2\mathbf{Q}\gamma - 2\mathbf{c} \quad (11)$$

where the prime denotes the transpose,  $\mathbf{Q}$  is a  $2n \times 2n$  matrix and  $\mathbf{c}$  is a vector of order  $2n$ .

Then the values of the  $2n$   $a$ 's and  $b$ 's which minimize the integral square error  $J_1$  can be obtained directly from

$$\gamma = \mathbf{Q}^{-1} \mathbf{c}. \quad (12)$$

Since all of the signals composing  $\mathbf{q}(t)$  and  $\hat{p}_{b_0}(t)$  can be obtained from the given  $x(t)$  and  $y(t)$  by means of the filtering defined by (9),  $\mathbf{Q}$  and  $\mathbf{c}$  can be computed either by analog or digital simulation from the previously estimated denominator  $D_{j-1}(s)$ .

The difference between the true error  $e(t)$  and the linear error  $e_1(t)$  is shown graphically in Fig. 1. If  $D_0(s)$  is assumed to be unity,  $e_1$  is simply the true error filtered by a device having  $D_1(s)$  as its transfer function. Therefore, in the first iteration  $e_1$  is analogous to the Kalman error used in sampled-data system identification [1], [2].

The iteration can thus be started either with a first estimate of  $D$  obtained from some other source, or with

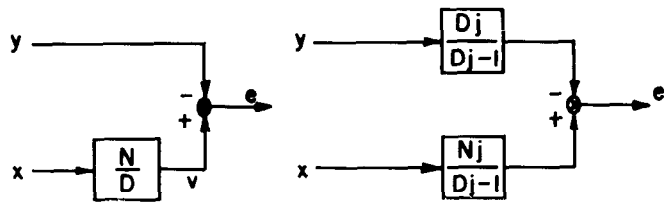


Fig. 1. True error and linear error compared.

the equivalent of a Kalman estimate. The computation of  $\gamma$  is repeated; if at some stage in the iteration  $D_i(s) \approx D_{i-1}(s)$  then  $e_1(t) \approx e(t)$  and  $J_1 \approx J$ . Such convergence cannot be guaranteed, but experimental evidence indicates that the procedure does converge in many practical problems, and does produce a significant reduction in true integral square error compared with the Kalman equivalent or various nonoptimal approximation techniques.<sup>2</sup>

MODE 2 ITERATION

When the Mode 1 iteration converges,  $J_1 = J$  but  $\text{grad } J_1 \neq \text{grad } J$ . Since it is the gradient that enters into the condition for a minimum, it might be expected that a procedure converging to  $\text{grad } J = 0$  would reduce the error still further.

Let a new signal variable  $\tilde{p}_{bi}(t)$  be defined by its transform

$$\tilde{P}_{bi}(s) = \frac{s^{-i}N_{i-1}(s)}{D_{i-1}^2(s)} X(s) = \frac{s^{-i}}{D_{i-1}(s)} V(s). \tag{13}$$

Then an approximation to (5) can be written

$$\begin{aligned} \frac{\partial J}{\partial a_i} &\approx 2 \int_0^T e_1(t)\hat{p}_{ai}(t) dt \\ \frac{\partial J}{\partial b_i} &\approx -2 \int_0^T e_1(t)\tilde{p}_{bi}(t) dt \end{aligned} \tag{14}$$

and since the right-hand side of (14) is linear in  $a_i, b_i$ , it can be set equal to zero and solved for the coefficients.

The computation is similar to that of Mode 1. If a signal vector  $\mathbf{p}(t)$  is defined by

$$\mathbf{p}(t) = \begin{bmatrix} \hat{p}_{a1}(t) \\ \vdots \\ \hat{p}_{an}(t) \\ \tilde{p}_{b1}(t) \\ \vdots \\ \tilde{p}_{bn}(t) \end{bmatrix} \tag{15}$$

the gradient of  $J$  with respect to  $\gamma$  becomes approximately

$$\begin{aligned} \text{grad } J &\approx 2 \left[ \int_0^T \mathbf{p}(t)\mathbf{q}'(t) dt \right] \gamma - 2 \int_0^T \hat{p}_{b0}(t)\mathbf{p}(t) dt \\ &= 2\tilde{\mathbf{Q}}\gamma - 2\tilde{\mathbf{c}}. \end{aligned} \tag{16}$$

It should be noted that this mode cannot be started from an initial estimate of  $G(s)$  which is either zero or unity. In the first case  $\tilde{p}_{bi} = 0$ , and in the second  $\tilde{p}_{bi} = \hat{p}_{ai}$ ; in either case  $\tilde{\mathbf{Q}}$  will be singular. At least one estimate must be made by a different method before Mode 2 can be used. As might be expected, the convergence of Mode 2 is more variable than that of Mode 1, but when it converges, it attains the true minimum.

Because Mode 1 uses the desired output  $y(t)$  as a substitute for the output  $v(t)$  from the optimal  $N/D$  [compare (9b) and (13)], it seems reasonable that in the early stages of iteration when  $v(t)$  is far from optimal, Mode 1 should give a better approximation to the true error surface, and therefore converge more rapidly. On the other hand, once the solution is in the neighborhood of the optimal point, the actual  $v$  is the best approximation to the optimal  $v$  and Mode 2 converges more rapidly. This general expectation has been borne out by experience in every case considered.

EFFECT OF NONZERO INITIAL CONDITIONS

In the foregoing analysis the energy stored in the network at  $t = 0$  was assumed to be zero. However, nonzero initial energy can be included in the problem, and the optimum values of various initial conditions computed simultaneously with the system parameters. This generalization is most conveniently made by adding a term to (3) to represent the output due to arbitrary initial conditions:

$$\begin{aligned} V(s) &= \frac{N(s)}{D(s)} X(s) + \frac{H(s)}{D(s)}, \\ H(s) &= h_n s^{-n} + \dots + h_1 s^{-1}. \end{aligned} \tag{17}$$

Then time functions  $r_i(t)$  and  $\hat{r}_i(t)$  can be defined in terms of their transforms as

$$\begin{aligned} \frac{\partial E}{\partial h_i} &= \frac{s^{-i}}{D(s)} = R_i(s) \\ \frac{\partial E_1}{\partial h_i} &= \frac{s^{-i}}{D_{i-1}(s)} = \hat{R}_i(s) \end{aligned} \tag{18}$$

and can be obtained through simulation as the impulse response of a system whose transfer function is  $s^{-i}/D$ .

<sup>2</sup> A similarly motivated iteration, proposed by Loeb [11] for fitting a ratio of polynomials to a known function in the Chebyshev sense, has been found strongly convergent in such problems [12].

TABLE I

Errors are Six-second Integrals of the Error Squared								
Model Order	Method	Initial Parameters	Error—Maximally Flat Approximation	Error—*Iteration 1	Error—Iteration 2	Error—Final	Delay Risetime (final)	
2	1	m.f.	0.15017	0.09382	0.09360	0.09382	1.1	
2	1	0		0.09543				
2	2	m.f.	0.15017	0.09299	0.09222	0.09171	1.1	
3	1	m.f.	0.11639	0.05644	0.05581	0.05640	1.9	
3	1	0		0.05733				
3	2	m.f.	0.11639	0.05728	0.05516	0.05459	1.9	
4	1	m.f.	0.09722	0.04040	0.03908	0.03972	2.6	
4	1	0		0.06767	0.03959	0.03927		
4	2	m.f.	0.09722	0.04143	0.03896	0.03801	2.6	
5	1	m.f.	0.08469	0.03206	0.03004	0.03045	3.3	
5	1	0		0.06711	0.03320	0.03001		
5	2	m.f.	0.08469	0.03258	0.03020	0.02886	3.3	
6	1	m.f.	0.07578	0.02860	0.02479	0.02462	4.3	
6	1	0		0.12988	0.07010	0.02885**		
6	2	m.f.	0.07579	0.03063	0.02450	0.02314	4.2	
7	2	m.f.	0.06910	0.02817	0.02133	0.01975	5.0	
8	1	m.f.	0.06387	0.03223	0.02034	0.01772	6.0	
8	2	m.f.	0.06387	0.02395	0.02186	0.01678***	5.8	

\* Error—Iteration 1 gives the error of the Kalman equivalent estimate in the case where I.C. = 0.

\*\* Third estimate.

\*\*\* Two iterations of Mode 2 started at the final coefficients obtained from Mode 1.

Then if the vectors  $\gamma$ ,  $\mathbf{p}$  and  $\mathbf{q}$  are augmented so that

$$\gamma = \begin{bmatrix} a_1 \\ \vdots \\ -b_n \\ h_1 \\ \vdots \\ h_n \end{bmatrix} \quad \mathbf{q}(t) = \begin{bmatrix} \hat{p}_{a_1}(t) \\ \vdots \\ \hat{p}_{b_n}(t) \\ \hat{r}_1(t) \\ \vdots \\ \hat{r}_n(t) \end{bmatrix} \quad \mathbf{p}(t) = \begin{bmatrix} \hat{p}_{a_1}(t) \\ \vdots \\ \hat{p}_{b_n}(t) \\ \hat{r}_1(t) \\ \vdots \\ \hat{r}_n(t) \end{bmatrix}$$

the remaining development for both Modes 1 and 2 follows without change. The values of  $h_i$  thus determined can be used to find the various initial energy storages of the final realization by conventional transform techniques.

### SIMULATION RESULTS

The two iterative techniques were applied to the problem of approximating the unit step response of a lumped-parameter transfer function to that of an ideal delay line. The delay time was one second, and the error squared was integrated from  $t = 0$  to  $t = 6$  seconds to form the quality criterion (the result approximates that which would be obtained over a semi-infinite interval, since the error is very small beyond  $t = 4$ ). A summary of the error criterion values resulting from various approximations is given in Table I. The initial parameter values from which the iterations were started were either 1) all zeros, in which case the first iteration yields the Kalman-equivalent estimate, or 2) coefficients obtained from the maximally flat approximation to a time delay [4], [5] taken from reference [5]. A commonly applied figure of merit, the delay-to-risetime ratio, is also given for the final set of parameters in each run (risetime is defined as the

interval during which the response is between 5 percent and 95 percent of its final value). As a standard for comparison, this ratio for the eighth-order maximally flat all-pole network is approximately 1.2.<sup>3</sup>

The step responses of the optimal third-, fifth-, and eighth-order networks from Mode 2 are shown in Figs. 2, 3, and 4. In Fig. 5 the response of the fifth-order Mode 1 network is shown for comparison. Comparison of Figs. 3 and 5 demonstrates the fact that although the reduction in error by the use of Mode 2 is only about 4 percent, the resulting step response is clearly superior.<sup>4</sup>

Reduction of integral square error by the iteration process is illustrated in Fig. 6. The errors shown for iteration 0 are those of the maximally flat approximation (off-scale in second- and third-order cases). The slightly erratic behavior for  $N = 7$  was found to be caused by too large a step size in the numerical integration routine. Note, however, that Mode 1 exhibits a relative minimum at the second or third iteration in nearly every case; such behavior seems to be typical of this procedure.

Figure 7 shows the manner in which the poles of the eighth-order network are shifted from their initial values to their final locations by the iterative procedures. Changes after the fourth iteration in Mode 1 (solid arrows) are too small to be shown effectively on this scale; it is clear that the requirement that  $e_1 \approx e$  is fulfilled after very few iterations. (The initial poles shown are the result of one iteration started from the maximally flat approximation.) The dashed arrows show the effect of two iterations in Mode 2, starting from the final Mode 1 poles.

<sup>3</sup> It has been brought to the authors' attention that the ratio for the Padé approximation to  $e^{-s}$  of order (7, 8) is 4.3.

<sup>4</sup> Transfer functions and poles and zeros of these systems are listed in Table II.

TABLE II

Fig. 2	$\frac{N}{D} = \frac{2.339s^2 - 14.49s + 38.15}{s^3 + 4.905s^2 + 23.88s + 37.85}$			
	Poles: $-1.40 \pm j4.00$ $-2.10$	Zeros: $3.10 \pm j2.59$		
Fig. 3	$\frac{N}{D} = \frac{3.804s^4 - 53.85s^3 + 573.5s^2 - 2903.4s + 6728}{1 + 10.557s^4 + 151.57s^3 + 875.1s^2 + 3768s + 6708}$			
	Poles: $-2.37 \pm j4.50$ $-1.46 \pm j9.36$ $-2.89$	Zeros: $3.95 \pm j2.61$ $3.13 \pm j8.31$		
Fig. 4	$\frac{N}{D} = \frac{-5.793s^7 + 14.28s^6 - 4197s^5 + 59117s^4 - 709650s^3 + 5126616s^2 - 24149837s + 51742153}{s^8 + 16.70s^7 + 685.0s^6 + 8279s^5 + 132583s^4 + 1030916s^3 + 7220545s^2 + 27077219s + 51849648}$			
	Poles: $-3.08 \pm j2.31$ $-1.75 \pm j12.9$	$-2.40 \pm j7.43$ $-1.12 \pm j18.4$	Zeros: $4.02 \pm j5.41$ $3.36 \pm j11.4$	$4.45$ $2.72 \pm j17.5$
Fig. 5	$\frac{N}{D} = \frac{3.62s^4 - 45.65s^3 + 502.35s^2 - 2497.5s + 5851.1}{s^5 + 7.704s^4 + 146.2s^3 + 705.36s^2 + 3423.5s + 5804.9}$			
	Poles: $-1.69 \pm j4.65$ $-1.00 \pm j10.1$ $-2.31$	Zeros: $3.65 \pm j2.63$ $2.65 \pm j8.53$		

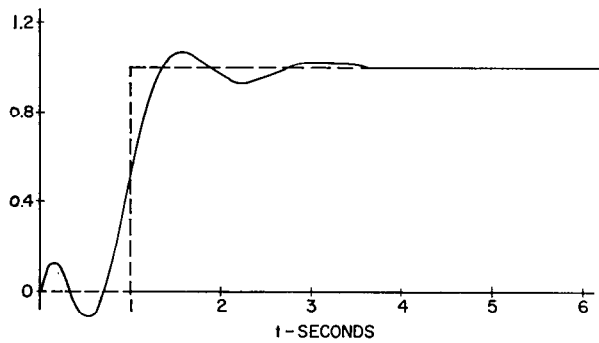


Fig. 2. Step response of third-order system (Mode 2).

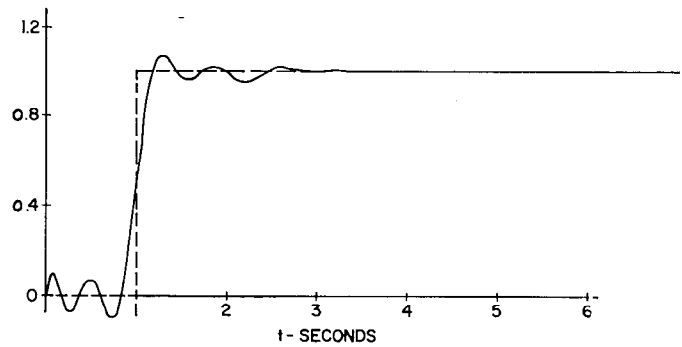


Fig. 3. Step response of fifth-order system (Mode 2).

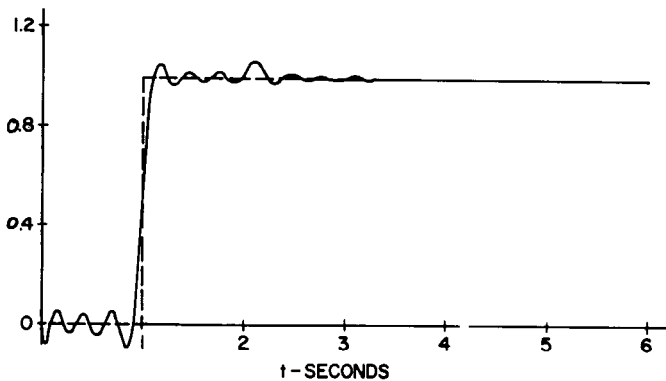


Fig. 4. Step response of eighth-order system (Mode 2).

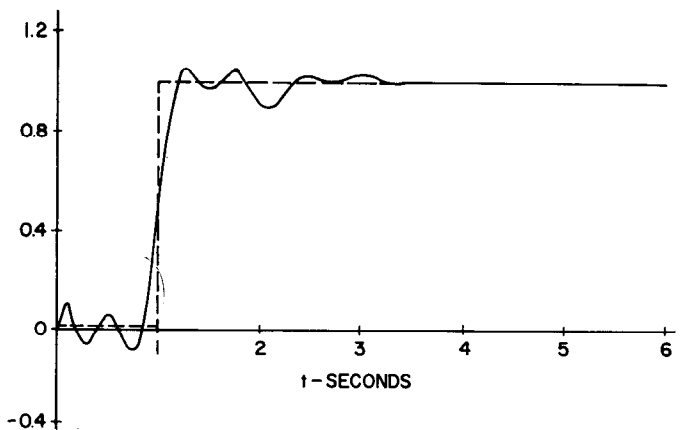


Fig. 5. Step response of suboptimal fifth-order system (Mode 1).

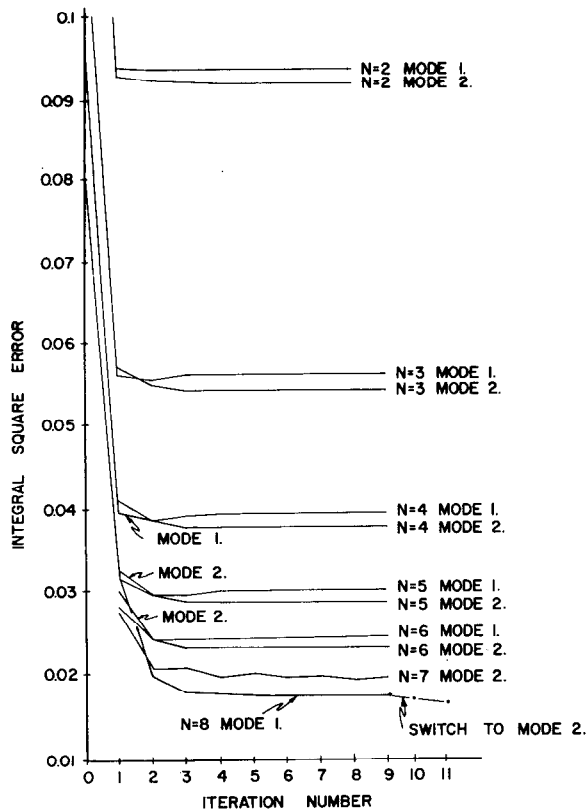


Fig. 6. Reduction of integral square error by iteration, starting from maximally flat approximation.

COMPARISON OF ITERATIVE METHODS

The method of steepest descent and other gradient techniques may be called first-order methods. They use only the first partial derivatives of the error criterion to determine the direction to be followed in parameter space; the step length along this line is determined either arbitrarily or by repeated gradient computation. Since for the problems considered here the gradient can always be obtained by filtering plus correlation, each iteration is relatively simple. However, the number of iterations is usually large, and it is at least intuitively obvious that the inclusion of additional information about the error surface should reduce the number of steps required.

The standard second-order technique is Newton's method, which requires both a gradient vector and a matrix of second partial derivatives at a point to construct the approximate quadratic surface defined by the three-term Taylor series. Since the equations for the optimum parameters are linear when the surface is quadratic, they can be solved by matrix inversion. A new quadratic surface is then constructed at this minimum and the process repeated.

For the problems considered in this paper, the necessary second-derivative matrix is

$$\frac{\partial^2 J}{\partial \gamma_i \partial \gamma_j} = 2 \int_0^T p_i(t)p_j(t) dt + 2 \int_0^T e(t) \frac{\partial^2 e(t)}{\partial \gamma_i \partial \gamma_j} dt \quad (19)$$

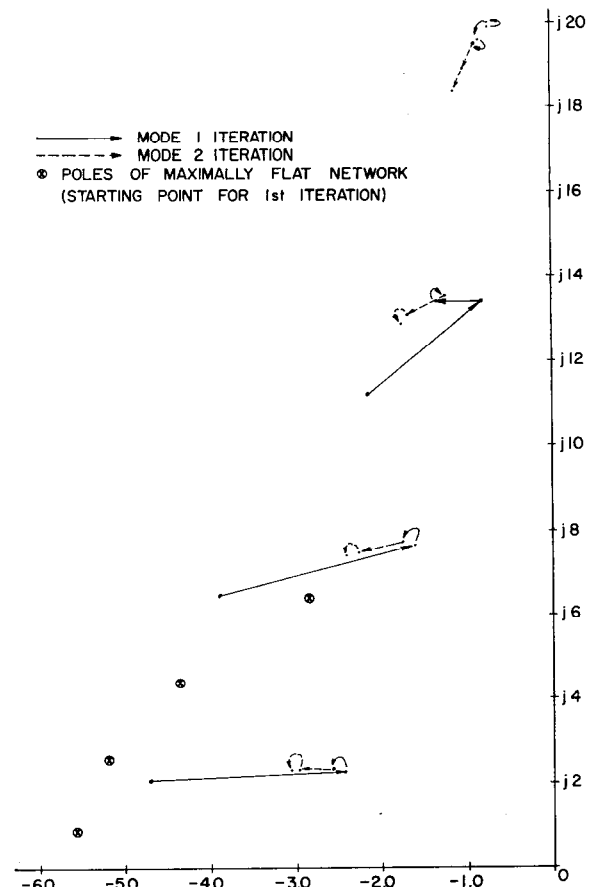


Fig. 7. Motion of eighth-order poles with iteration.

since

$$p_i(t) = \frac{\partial e(t)}{\partial \gamma_i}$$

It can be seen [e.g., by differentiating (13)] that the second term of (19) can also be computed by a filtering and correlation process, so that the Newton method could be used directly. However,  $4n^2$  filtering operations would be required and each iteration made more complex. This approach has not been tried but might prove useful in certain cases. The (Newton's) matrix equation to be solved for  $\gamma$  then becomes

$$\gamma = \left[ \frac{\partial^2 J}{\partial \gamma_i \partial \gamma_j} \right]^{-1} 2 \int_0^T e(t) \mathbf{p}(t) dt \quad (20)$$

where the expression in square brackets is the matrix whose components are defined by (19).

A simplified iterative method can be devised by simply omitting the second term of (19). Since this term would be zero if  $e$  were a linear function of  $\gamma$  (the error surface were quadratic), the procedure is identical to Newton's in this case. It has been called the Gauss-Newton method and explored by a number of authors [6], [7]. Its principal attraction lies in the fact that it is a second-order method requiring only the gradient vector  $\mathbf{p}(t)$ ; it can thus be

employed in any problem for which a gradient computer is available. For example, the method of quasi-linearization [8], [9] is an iterative procedure based on gradient computation alone; if it is used with noisy measurements to find the set of parameters which minimizes a mean square error between observation and prediction (as suggested in [9]) the resulting equations can be written in the form of (19) and (20) with the second term of (19) omitted. Unfortunately, computational simplicity is obtained at the expense of convergence in many problems. Reference [7] gives a necessary and sufficient condition for the existence of any region of convergence at all, and also gives examples requiring another iterative method to bring the solution within a convergent region.

The methods presented in this paper, both Modes 1 and 2, are also second-order methods in that they define a quadratic error surface at each successive solution. However, the surface is obtained, not from the Taylor series for the true surface, but by defining a new error,  $E_1$ , [see (8)]. This procedure has two advantages as compared to the other second-order methods: 1) it is computationally simpler than any other except the Gauss-Newton methods; 2) for problems in which the minimum value of the error criterion is zero,  $E_1 = E$  and the first iteration (Kalman equivalent) gives the exact solution. (These no-noise problems include most of those usually cited in discussions of identification; all of the illustrative examples of [7] and [9] belong to this class.)

When zero error is impossible, general statements about convergence are difficult to verify. Mode 1 has proved convergent for arbitrary starting points in all of the problems discussed here and in many others, and it has always been possible to find a region in which Mode 2 is convergent by the use of Mode 1 or a mixed mode in which  $\hat{p}$  is gradually replaced by  $\bar{p}$ .

Although these methods do not appear applicable in their present form to the most general problem for which the Newton or gradient methods are suited, they can be directly applied to many problems besides time-domain synthesis, including the fitting to measured data of nonlinear differential equations in which the unknown parameters appear linearly. Other fields in which additional work is needed are prediction and spectrum identification, for which the Kalman-equivalent problem is singular; it seems possible that alternate computation of  $N$  and  $D$  separately will permit the application of iteration to these problems.

One objection to all of these methods is the inefficiency of the digital computer in the integration of differential equations; because each iteration involves filtering and correlation, the computation time is much longer than is needed for sampled-data problems [2]. (The eighth-order delay synthesis required nearly two minutes per iteration on the IBM 7094.) One way of eliminating this difficulty is the transformation of the continuous problem to another space where the same coefficients appear as the solution to a sampled-data problem. This procedure has been used in synthesizing a delay line by means of

Laguerre coefficients [10] and will be discussed under more general conditions in a later report. The advantage of such an approach is the confining of all integration to a single step (in [10] this step is the one in which the Laguerre coefficients are determined), while the iterative procedures necessary to obtain a rational transfer function involve only difference equations.

## CONCLUSION

A method has been proposed which is applicable to the general problem of optimal time-domain synthesis, defined as the determination of the rational transfer function of given order whose output when subjected to a known input function best approximates, in an integral square sense, a given desired time function. Since the problem of linear system identification from operating records may be posed in similar fashion, the technique is applicable to both.

As an illustration of the practicality of the iterative approach, lumped delay lines of orders from second through eighth have been synthesized; they are optimal (have minimum integral square error) for step inputs. Effective convergence in such problems appears to occur within three to five iterations. The method appears to have advantages over other iterative procedures for this class of problems, requiring fewer iterations for convergence and simpler computations for each iteration. The proposed technique is easily programmed for the digital computer, but requires longer computing times than the analogous sampled-data identification problem.

## REFERENCES

- [1] R. E. Kalman, "Design of a self-optimizing system," *Trans. ASME*, vol. 80, pp. 468-478, February 1958.
- [2] K. Steiglitz and L. E. McBride, "A technique for the identification of linear systems," *IEEE Trans. on Automatic Control (Short Papers)*, vol. AC-10, pp. 461-464, October 1965.
- [3] P. J. Woodrow, "Methods for locating the extrema of multivariate functions," Control Sys. Lab., Princeton University, Princeton, N. J., Tech. Rept. 4, November 1962.
- [4] L. Storch, "Synthesis of constant-time-delay ladder networks using Bessel polynomials," *Proc. IRE*, vol. 42, pp. 1666-1675, November 1954.
- [5] M. E. Van Valkenburg, *Introduction to Modern Network Synthesis*. New York: Wiley, 1960.
- [6] H. O. Hartley, "The modified Gauss-Newton method for the fitting of nonlinear regression functions by least squares," *Technometrics*, vol. 3, pp. 269-280, May 1961.
- [7] G. A. Bekey and R. B. McGhee, "Gradient methods for the optimization of dynamic system parameters by hybrid computation," in *Computing Methods in Optimization Problems*, A. V. Balakrishnan and L. V. Neustadt, Eds. New York: Academic, 1964, pp. 305-327.
- [8] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton, N. J.: Princeton University Press, 1961.
- [9] K. S. Prasanna Kumar and R. Sridhar, "On the identification of control systems by the quasi-linearization method," *IEEE Trans. on Automatic Control*, vol. AC-9, pp. 151-154, April 1964.
- [10] K. Steiglitz, "Rational transform approximation via the Laguerre spectrum," *J. Franklin Inst.*, vol. 280, pp. 387-394, November 1965.
- [11] H. L. Loeb, "A note on Rational Function Approximation," Convair Astronautics, Appl. Math. Ser. no. 27, September 1959.
- [12] E. W. Cheney and T. H. Southard, "A survey of methods for rational approximation," *SIAM Review*, vol. 5, pp. 219-231, July 1963.
- [13] W. H. Kautz, "Network synthesis for specified transient response," M.I.T. Research Lab. Electronics, Cambridge, Mass., Tech. Rept. 209, April 23, 1952.